NL 000711
20 dec

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 178 027 B1**

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication of patent specification: **25.09.91**

(51) Int. Cl.5: **H03M 5/14**, G11B 20/14, H04L 25/49

(21) Application number: **85201620.3**

(22) Date of filing: **07.10.85**

(54) **Method of transmitting information, encoding devide for use in the method, and decoding device for use in the method.**

(30) Priority: **10.10.84 NL 8403078**

(43) Date of publication of application:
16.04.86 Bulletin 86/16

(45) Publication of the grant of the patent:
25.09.91 Bulletin 91/39

(84) Designated Contracting States:
DE FR GB SE

(56) References cited:
DE-A- 1 963 945
FR-A- 2 469 046
FR-A- 2 469 047
GB-A- 2 094 107
US-A- 4 261 019

IBM TECHNICAL DISCLOSURE BULLETIN, vol. 24, no. 8, January 1982, pages 4087-4089, New York, US; P.A. FRANASZEK et al.: "DC balanced run-length limited code"

IBM TECHNICAL DISCLOSURE BULLETIN, vol. 24, no. 6, November 1981, pages 2759-2762, New York, US; J.A. ROONEY: "DC limited recording code and implementation"

S.M.P.T.E. JOURNAL, vol. 92, no. 9, September 1983, pages 918-922, Scarsdale, New York, US; H. YOSHIDA et al.: "8-9 block code: A DE-free channel code for digital magnetic recording"

(73) Proprietor: **N.V. Philips' Gloeilampenfabrieken**
Groenewoudseweg 1
NL-5621 BA Eindhoven(NL)

(72) Inventor: **Schouhamer Immink, Kornelis Antonie**
c/o INT. OCTROOIBUREAU B.V. Prof. Holstlaan 6
NL-5656 AA Eindhoven(NL)

(74) Representative: **Beckers, Hubertus Franciscus Maria**
INTERNATIONAAL OCTROOIBUREAU B.V.
Prof. Holstlaan 6
NL-5656 AA Eindhoven(NL)

EP 0 178 027 B1

Description

The invention relates to a method of transmitting information, in which prior to transmission m-bit information words are converted into n-bit code words and after transmission said n-bits code words are reconverted into m-bit information words, where m < n and m and n are positive integers, and in which in order to obtain a d.c. free transmission the possible code words have been selected in such a way that the running sum

$$\sum_{j=1}^{i} x_j$$

over the bits of the generated code words, where $x_j = \pm 1$ is the value of the j^{th} bit remains restricted regardless of the sequence of information words.

The invention also relates to an encoding device for use in the method.

Moreover, the invention relates to a decoding device for use in the method.

Such methods and devices are employed in order to optimise the signal spectrum for transmission. For example, in the case of transmission via a magnetic tape, but also in the case of transmission via other media such as cables, space or optical record carriers, it is desirable to have a d.c. free signal and a spectrum with a minimal low-frequency content in order to obtain room for various control signals such as tracking signals in recording. It is known, inter alia from GB-PS 1,540,617, that a d.c. free code is obtained if the running sum over the consecutively generated bits remains restricted (a value opposite to the value of a logic "1" being assigned to a logic "0"), which means that there will not be a series of input words which gives rise to a series of code words whose running sum increases towards infinity. The simplest method of guaranteeing this is to use for the code words only those words which have a disparity (which is the sum over the bit value of the code word, i.e. the variation of the running sum caused by this code word) equal to zero, i.e. to use only code words comprising as many ones as zeros. An example of this is the "biphase" code in accordance with the table in Fig. 2. Another simple method, whith results in a larger number of possible code words and which is therefore more efficient, is described in said Britisch Patent Specification. In addition to code words of zero disparity, code words of a specific non-zero disparity are permissible, both a code word of positive disparity and a code word of a disparity opposite thereto being assigned to the relevant input words (for example +2 and -2), from which two code words a choice is made which depends on the digital sum value (running sum) over the preceding code words, in such a way that absolute value of this digital sum value remains restricted. The simplest solution is to select one set and to generate the associated words of the other set by inverting the words of said one set.

The invention aims at providing a method of the kind defined in the opening paragraph, and an encoding device and a decoding device for use in this method, in which the spectrum of the code words generated in the case of a random supply of information words has a low-frequency signal content which is smaller than in the case of the known method and devices.

The method in accordance with the invention is therefore characterized in that in addition the possible code words are selected in such a way that the running sum

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

over all the preceding sum values over the bits of the generated code words also remains restricted.

The invention is based on the recognition of the fact that by limiting the sum

$$\sum_{k=1}^{I} \sum_{j=1}^{k} x_j$$

the second derivative S'' (ω) of the energy spectrum S(ω) is zero for $\omega = 0$. which means a reduction of the low-frequency content of the spectrum.

The method in accordance with the invention may be characterized further in that to a first group of information words code words are assigned which comply with

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j = 0$$

for every n-bit code word, to
a first group of information words first and second code words are assigned for which the sum

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j$$

has a
value +a and -b, respectively, and when an information word of said first group appears, a choice is from the two possible code words as a function of the running sum

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

over all the preceding code words, in such a way that said running sum is minimized,

The encoding device in accordance with the invention may be characterized in that the parameters a and b are equal.

The decoding device in accordance with the invention may be characterized by a conversion circuit for converting the m-bit information words into n-bit code words in such a way that the running sum

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

over all the preceding sum
values over the bits of the generated code words remains restricted.

The invention will now be described in more detail, by way of example, with reference to the accompanying drawings, in which:

Fig. 1 shows a known device using a method of transmitting information,
Fig. 2 shows a conversion table used in a known encoding/decoding method,
Fig. 3 shows a table for an encoding/decoding method in accordance with the invention,
Fig. 4 shows a table to illustrate the efficiency R of a group of encoding/decoding methods in accordance with the invention,
Fig. 5 is a table illustrating the effiency of another group of encoding/decoding methods in accordance with the invention,
Fig. 6 is a table relating to a preferred version of an encoding/decoding method in accordance with the invention,
Fig. 7 is a table to explain the use of the code in accordance with the table of Fig. 6,
Fig. 8 shows a device using the preferred version of the method in accordance with the invention, and
Fig. 9 is a graph illustrating the effect of the invention and showing the energy spectrum obtained when the codes of the tables of Figs, 2 and 6 are used.

Fig. 1 shows a device employing a system of encoding and decoding digital data. The device comprises an input 1 for receiving serial input data and (unless the data is presented in parallel form) applying said data to a series-to-parallel converter 2 for arranging the input data in 2-bit parallel words in the present example. These 2-bit words are applied to an encoding circuit 3, for example in the form of a look-up table or a gate circuit, which encoding circuit, in the present example, generates a 4-bit output word for every input word in conformity with the rules laid down by means of this encoding circuit. By means of a parallel-to-series converter 4 these 4-bit words are converted into a serial data sequence, which is, for example, recorded on a magnetic tape by means of a conventional analog magnetic-tape recorder 6. It is then possible to record, for example, a plurality of parallel tracks. Synchronisation is effected by means of clock signals which are derived from the input signal by a clock-signal generator circuit 5, for example by means of a phase-locked loop.

In principle, decoding may be effected by means of a similar circuit arrangement through which the

signals are fed in reverse sequence. The signal from the tape recorder 6 is arranged in groups of 4-bit words by a series-to-parallel converter 7 (unless the data is available as 4-bit words). These 4-bit words are converted into 2-bit words with a decoder circuit 8 in conformity with rules which are complementary to those used for encoding, and are subsequently converted into a serial data stream on output 10 by means of a parallel-to-series converter 9. Again, this process is synchronised with clock signals which are derived, by means of the clock-signal generator circuit 13, from the signal which is obtained from the recorder 6 and which appears on input 12 of the series-to-parallel converter 7.

Conversion in the encoding circuit 3 is effected inter alia in order to obtain a signal spectrum which is suitable for transmission. For example, in the case of transmission via a magnetic tape, but also in the case of transmission via other media such as cables, space or optical record carriers, it is desirable to have a d.c. free signal and a spectrum with a minimal low-frequency content to enable various control signals, such as tracking signals in the case of recording, to be accommodated. For example from GB-PS 1,540,617 it is known that a d.c. free code is obtained when the running sum over the consecutively generated bits is limited (a value opposite to the value of a logic "1" being assigned to a logic "0"), which means that there is no series of input words which gives rise to a series of code words whose running sum grows towards infinity. The simplest method of guaranteeing this is to use for the code words only those words whose disparity (which is the sum over the bit value of the code word, i.e. the variation of the running sum caused by this code word) is zero, so only code words comprising as many ones as zeros. An example of this is the "Biphase" code given in the form of a table in Fig. 2. Another simple method which allows more possible code words and is therefore more efficient, is described in said Britisch Patent Specification. Apart from code words with zero disparity, code words with a specific non-zero disparity are also permissible, both a code word of positive disparity and a code word of opposite disparity being assigned to a code word (for example +2 and -2) and a choice being made from these two words depending on the digital sum value (running sum) over the preceding code words, in such a way that the absolute value of this digital sum value remains restricted. The simplest way is to select one set and to generate the associated words of the other set by inverting the words of the first-mentioned set.

In the device shown in Fig. 1 this can be achieved by determining the digital sum value over all the preceding words with an up/down counter 14 which counts down upon receipt of every logic zero and which counts up upon receipt of every logic one and by generating as a function thereof a logic signal S_o/S_1 which indicates whether this digital sum value has a high (S_1) or a low (S_o) value of two possible values. If the value of S_o is low, the next input word is converted into a word of zero or +2 disparity in conformity with the obtaining rules, so that the digital sum value remains S_o or becomes S_1 ($S_1 = S_o + 2$) and, if the value of S_1 is high, this input word is converted into a word of zero or -2 disparity, so that the digital sum value remains S_1 or becomes S_o ($S_o = S_1-2$).

During decoding the digital sum value of all the preceding words is determined by means of the up/down counter 15 in order to determine as a function thereof whether during encoding a word of 0 or +2 disparity or a word of 0 or -2 disparity has been selected as the next code word. The decoding circuit 8 is controlled as a function of this. Both the encoding and the decoding circuit therefore comprise a set of code words S_o, obtained by means of rules or tables, which is valid if the digital sum value of all the preceding code words is S_o and a set S_1 which is valid if the digital sum value of all the preceding words is S_1.

In accordance with the afore-mentioned Patent Specification one set S_1 can be derived simply from the other set S_o when the words of zero disparity in both sets are selected to be identical and the words of -2 disparity are selected to be complementary to the words of +2 disparity. The Applicant's Patent Application 8402444 (PHN 11.127) describes an improvement to this method.

Mathematically, the above may be defined as follows:

$$S(\omega=0) = 0 \text{ if } Z_o(i) = \sum_{j=1}^{i} x_j \qquad (1)$$

remains limited as i increases, where $S(\omega)$ is the energy spectrum, $Z_0(i)$ is the sum over all i preceding bits x_j, with $x_j = \epsilon\{+1, -1\}$, j the sequence number of the bits x_j, and ω the angular frequency. In the said simplest case this requirement is therefore met if: for each

$$Z_0(n) = \sum_{j=1}^{n} x_j = 0 \qquad (2)$$

code word, i.e. for each (n-bit) code word the disparity is zero.

Further, in accordance with the invention:

$$S''(\omega=0) = 0 \text{ if } z_1(i) = \sum_{k=1}^{i} z_0(k) = \sum_{k=1}^{i} \sum_{j=1}^{k} x_j \qquad (3)$$

remains limited as i increases, which means that the second derivative S'' (ω) of the energy spectrum S (ω) (the odd derivatives of S(ω) are always zero for $\omega = 0$) are zero for $\omega = 0$ when the sum z_1 (i) over all i preceding sums of k bits is restricted. In practice, the second derivative S'' (ω) being zero means that the low-frequency content of the spectrum is even smaller than in the case in which only S($\omega = 0$) is zero, which may improve the usefulness of the code.

A further reduction of the low-frequency content can be obtained if:

$$z_2(i) = \sum_{k=1}^{i} z_1(k) = \sum_{k=1}^{i} \sum_{l=1}^{k} z_0(l) = \sum_{k=1}^{i} \sum_{l=1}^{k} \sum_{j=1}^{l} x_j \qquad (4)$$

remains restricted as i increases.

Similarly, this requirement may be imposed on $z_3(i)$, $z_4(i)$, $z_5(i)$....$z_p(i)$....., i.e. a summation every time. In general, it may be stated that

$$S^{2p}(\omega=0) = 0 \text{ if } z_p(i) = \sum_{k=1}^{i} z_{p-1}(k) \qquad (5)$$

remains restricted, where S^{2p} (ω) is the $2p^{th}$ derivative of the energy spectrum S(ω), $z_p(i)$ is the p^{th} order sum, and z_{p-1} is the $(p-1)^{th}$ order sum. In practice, these further requirements and the requirement for z_2 generally do not lead to efficient codes, so that only requirement (3) will be of practical significance.

Requirement (3), like requirement (1), can be met by imposing this requirement on every code word; this means that only those code words should be selected which for every n-bit code word comply with:

$$z_1(n) = \sum_{k=1}^{n} \sum_{j=1}^{k} x_j = 0 \qquad (6)$$

This requirement is met if each code word complies with:

$$\sum_{j=1}^{n} j x_j = 0 \qquad (7)$$

This requirement means that for each code word the sum over the product of the bit value and the sequence number of this bit for every word is zero. An example of a group of (2) code words which complies with requirement (6) is given in the table of Fig. 3, which gives the two possible 4-bit code words which meet requirement (6) and the associated 1-bit input words. The efficiency of such a conversion is 0.25 because the ratio between the input bit rate and the output bit rate is 0.25.

Similarly, the simplest way of meeting requirement (5) is to impose this requirement on every word. In general, this is met if every code word complies with:

$$\sum_{j=1}^{n} j^{p} x_j = 0 \qquad (7)$$

where p is the order of the summation $z_p(i)$.

Another method of simply meeting requirement (3) is to select for each input word two code words for which the parameters $z_1(n)$ have non-zero values, as the case may be together with code words whose parameters $z_1(n)$ are zero, and to select one of the two code words as a function of the logic sum $z_1(i)$ over all i/n preceding n-bit words, in such a way that this running sum $z_1(i)$ remains restricted. In addition, to meet requirement (1) that the running sum $z_o(i)$ (= digital sum value) remains restricted, any appropriate method may be adopted. A suitable choice is the choice for code words of non-zero disparity ($z_o(n) = 0$). If in addition those code words are selected whose sum $z_1(n)$ is also zero, the table in Fig. 4 gives the number M of possible code words which can be formed with n-bit words and the attainable efficiency $R = {}^2\log(M)/n$. It is evident that for a more or less acceptable efficiency (for example, $R \geq 0.5$) comparatively long code words of 16 or more bits have to be selected (all the code words should have a length equal to a multiple of 4 bits to meet the requirement that both $z_o(n)$ and $z_1(n)$ should be zero).

More efficient codes appear to be possible by maintaining the requirement $z_o(n) = 0$ and admitting more values of $z_1(n)$. The table of Fig. 5 shows the code efficiency R, in which the first column gives the number of bits n of the code words and above each column the number of possible values r is given which the parameter $z_1(i)$ may assume at the end of each code word.

The table in Fig. 6 gives an example of a code where $n = 4$ and $r = 4$ (Fig. 5), which code is equivalent to the said biphase code (Fig. 2) as regards the efficiency R and the length of the code words. The first column of the table shows the 2-bit input word, the second column gives the associated 4-bit output word when the parameter $z_1(i)$ over all the preceding words is smaller than zero, the third column gives the value of $z_1(n)$ of this word, the fourth column gives the output word when the parameter $z_1(i)$ over all the preceding words is greater than zero, and the fifth column gives the value of $z_1(n)$ over this word. At the end of every word $z_1(i)$ then always has the four possible values +3, +1, -1 and -3.

Fig. 7 shows the table of Fig. 6 in more detailed form in order to translate the encoding rules into an encoding gate circuit.

In this example the parameter $z_1(n)$ may have values of +4 and -4, respectively, +2 and -2, respectively, and zero. The first column gives the 2-bit input word. The second column gives the value of the running sum $z_1(i)$ over all the preceding words in 2-bit notation and decimal notation, the third column gives the output code word to be selected, the fourth column the value of the sum $z_1(n)$ of this code word, and the fifth column gives the new value of the running sum $z_1(i + n)$ in two-bit notation and decimal notation.

As is shown in the table, the input words 00 and 01 are converted into the words 1001 and 0110, respectively, regardless of the running sum $z_1(i)$, which words each exhibit a sum $z_1(n)$ equal to zero, so that $z_1(i + n)$ remains equal to $z_1(i)$. The input word 10 is converted into an output word 0101 with $z_1(n) = -2$ when the sum $z_1(i)$ was equal to 00 or 01 (+3 and +1, respectively) and into an output word 1010 with $z_1(n) = +2$ when the sum $z_1(i)$ was 10 or 11 (-1 and -3, respectively). Similarly, the input word 11 is converted into an output word 0011 with $z_1(n) = -4$ when the sum $z_1(i)$ was 00 or 01 and into an output word 1100 with $z_1(n) = +4$ when this sum $z_1(i)$ was 10 or 11.

The code of Fig. 7 can be decoded without determining the parameter $z_1(i)$, because all the code words can be re-converted unambiguously into the original data words. This is even possible with only 3 of the 4 bits. This is because the code words have been selected with the requirement that $z_o(n) = 0$, i.e. code words comprising as many ones as zeros. Therefore, the information is defined by 3 of the 4 bits; the fourth bit merely ensures that $z_o(n) = 0$.

Fig. 8 shows an example of encoding and decoding in conformity with the table of Fig. 7. The device corresponds to that of Fig. 1 except for the sections 14 and 15 which are not necessary because there are only code words with $z_o(n) = 0$. In the encoding device 3 the input bits a_2 and a_3 and the parameter $z_1(i)$ representing bits a_0 and a_1 are converted into output bits b_2, b_3, b_4 and b_5 and the parameter $z_1(i + 4)$ representing the bits b_o and b_1 in conformity with the indication in the second row of the table in Fig. 7.

The bits a_3, a_2, a_1 and a_0 are applied to inverting and non-inverting inputs of AND-gates A_0 to A_{10} in which they are combined in such a way that on the outputs of these AND-gates A_o to A_{10} the bits c_o to c_{10} appear in conformity with the following logic equations:

$c_o = a_o.a_2$

$c_1 = a_o.\overline{a_3}$

$c_2 = a_1.\bar{a}_3$
$c_3 = \bar{a}_o.\bar{a}_1.\bar{a}_2.a_3$
$c_4 = a_o.\bar{a}_1.\bar{a}_2.a_3$
$c_5 = \bar{a}_o.a_1.\bar{a}_2.a_3$
$c_6 = a_o.a_1.\bar{a}_2.a_3$
$c_7 = \bar{a}_1.a_2.a_3$
$c_8 = \bar{a}_2.\bar{a}_3$
$c_9 = a_2.\overline{a_3}$
$c_{10} = a_1.a_2.a_3$

These bits c_o to c_{10} are subsequently combined to give bits b_o to b_5 by OR-gates 0_o to 0_5 in accordance with the following logic equations:

$b_o = c_o + c_1 + c_3 + c_5$
$b_1 = c_2 + c_4 + c_6 + c_7$
$b_2 = c_3 + c_4 + c_7 + c_8$
$b_3 = c_5 + c_6 + c_7 + c_9$
$b_4 = c_3 + c_4 + c_9 + c_{10}$
$b_5 = c_5 + c_6 + c_8 + c_{10}$

The bits b_o and b_1 represent the new value of the parameter $z_1 = z_1 (i + 4)$ and are applied to the flip-flops FF_1 and FF_2 where they are latched until the next input word a_2, a_3 is converted.

In the decoding circuit 8 the bits b_3, b_4 and b_5 are combined to form bits d_1 to d_4 in AND-gates A_{11} to A_{15} in accordance with the following logic equations:

$d_o = \overline{b_5}.\overline{b_4}$
$d_1 = b_5.b_4$
$d_2 = \overline{b_5}.\overline{b_3}$
$d_3 = b_5.b_3$
$d_4 = \overline{b_5}.b_3$

By means of OR-gates 0_6 and 0_7 these bits d_o to d_4 are combined to form the output bits a_3 and a_2 in conformity with the logic equations

$a_3 = d_o + d_1 + d_2 + d_3$
$a_2 = d_1 + d_4$

To illustrate the effect of the encoding method in accordance with the invention Fig. 9 shows the energy spectrum $S(\omega)$ in dB as a function of the relative angular frequency ω/ω_o (ω_o is the channel bit frequency, which spectrum is symmetrical about $\omega/\omega_o = 0.5$) on a logarithmic scale. Curve I represents the spectrum when the biphase code of Fig. 2 is used an curve II gives the spectrum when the code of Fig. 7 is used. It can be seen that for low frequencies the code in accordance with the invention produces a smaller signal, as was required.

Claims

1. A method of transmitting information, in which prior to transmission m-bit information words are converted into n-bit code words and after transmission said n-bits code words are re-converted into m-bit information words, where $m < n$ and m and n are positive integers, and in which in order to obtain a d.c. free transmission the possible code words have been selected in such a way that the running sum

$$\sum_{j=1}^{i} x_j$$

over the bits of the generated code words, where
$x_j = \pm 1$ is the value of the j^{th} bit and i is a positive integer, remans restricted regardless of the sequence of information words, characterized in that in addition the possible code words are selected in such a way that the running sum

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

over all the preceding sum values (K is a positive integer) over the bits of the generated code words also remains restricted.

2. A method as claimed in Claim 1, characterized in that to a first group of information words code words are assigned which comply with

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j = 0$$

for every n-bit code word.

3. A method as claimed in Claim 1, characterized in that to a first group of information words first and second code words are assigned for which the sum

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j$$

has a
value +a and -b, respectively, and when an information word of said first group appears, a choice is made from the two possible code words as a function of the running sum

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

over all the preceding code words, in such a way that said running sum is minimized.

4. A method as claimed in Claim 2, characterized in that to a second group of information words first and second code words are assigned for which the sum

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j$$

has a value +a and -b respectivejy, and when an information word of said second group appears a choice is made from the two possible code words as a function of the running sum

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

over all the preceeding code words, in such a way that said running sum is minimized.

5. A method as claimed in Claim 3 or 4, characterized in that the parameters a and b are equal.

6. An encoding device for use in the method as claimed in any one or any of the preceding Claims,

characterized by a conversion circuit for converting the m-bit information words into n-bit code words in such a way that the running sum

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

over
all the preceding sum values over the bits of the generated code words remains restricted.

7. An decoding device for use in the method as claimed in any one or any of the following Claims, characterized by a conversion circuit for converting n-bit code words into m-bit information words, which conversion circuit is adapted to receive a sequence of code words whose running sum

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

over all the preceeding sum values over the bits of the code words received remains restricted.

Revendications

1. Procédé de transmission d'information, dans lequel, avant la transmission, des mots d'information de m bits sont convertis en mots de code de n bits et, après la transmission, ces mots de code de n bits sont reconvertis en mots d'information de m bits, où $m < n$ et m et n sont des nombres entiers positifs, et dans lequel, afin d'obtenir une transmission exempte de composante continue, les mots de code possibles ont été sélectionnés d'une manière telle que la somme courante

$$\sum_{j=1}^{i} x_j$$

sur les bits des mots de code générés où $x_j = \pm 1$, est la valeur du j-ième bit et i est un nombre entier positif, reste limitée, indépendamment de la séquence de mots d'information, caractérisé en ce qu'en outre, les mots de code possibles sont sélectionnés d'une manière telle que la somme courante

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

sur toutes les valeurs de somme précédentes (k est un nombre entier positif) sur les bits des mots de code générés reste également limitée.

2. Procédé suivant la revendication 1, caractérisé en ce qu'à un premier groupe de mots d'information sont attribués des mots de code qui sont conformes à

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j = 0$$

pour chaque mot de code de n bits.

3. Procédé suivant la revendication 1, caractérisé en ce qu'à un premier groupe de mots d'information

sont attribués des premiers et seconds mots de code pour lesquels la somme

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j$$

a respectivement une valeur +a et -b, et lorsqu'un mot d'information du premier groupe apparaît, un choix est effectué parmi les deux mots de code possibles en fonction de la somme courante

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

sur tous les mots de code précédents, d'une manière telle que la somme courante soit réduite au minimum.

4. Procédé suivant la revendication 2, caractérisé en ce qu'à un second groupe de mots d'information sont attribués des premiers et seconds mots de code pour lesquels la somme

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j$$

a respectivement une valeur +a et -b, et lorsqu'un mot d'information du second groupe apparaît, un choix est effectué parmi les deux mots de code possibles en fonction de la somme courante

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

sur tous les mots de code précédents, d'une manière telle que la somme courante soit réduite au minimum.

5. Procédé suivant la revendication 3 ou 4, caractérisé en ce que les paramètres a et b sont égaux.

6. Dispositif de codage à utiliser dans le procédé suivant l'une quelconque des revendications précédentes, caractérisé par un circuit de conversion pour convertir les mots d'information de m bits en mots de code de n bits d'une manière telle que la somme courante

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

sur toutes les valeurs de somme précédentes sur les bits des mots de code générés reste limitée.

7. Dispositif de décodage à utiliser dans le procédé suivant l'une quelconque des revendications précédentes, caractérisé par un circuit de conversion pour convertir des mots de code de n bits en mots d'information de m bits, ce circuit de conversion étant à même de recevoir une séquence de mots de code dont la somme courante

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

sur toutes les valeurs de somme précédentes sur les bits des mots de code recus reste limitée.

Patentansprüche

1. Verfahren zum Übertragen von Information, bei dem vor der Übertragung m-Bit-Informationswörter in n-Bit-Codewörter umgesetzt und nach der Übertragung die n-Bit-Codewörter in m-Bit-Informationswörter rückumgesetzt werden, worin m < n ist und m und n positive ganze Zahlen sind, und bei dem zum Erhalten einer gleichstromfreien Übertragung die möglichen Codewörter derart gewählt sind, daß die laufende Summe

$$\sum_{j=1}^{i} x_j$$

über die Bits der erzeugten Codewörter, worin $X_j = \pm 1$ der Wert des j. Bit und i eine positive ganze Zahl sind, ungeachtet der Reihenfolge der
Informationswörter beschränkt bleibt, dadurch gekennzeichnet, daß zusätzlich die möglichen Codewörter derart gewählt werden, daß
die laufende Summe

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

über sämtliche vorangehende Summenwerte (k ist eine positive ganze Zahl) über die Bits der erzeugten Codewörter ebenfalls beschränkt bleibt.

2. Verfahren nach Anspruch 1,
dadurch gekennzeichnet, daß einer ersten Gruppe von Informationswörtern Codewörter zugeordnet sind, die für jedes n-Bit-Codewort übereinstimmen mit

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j = 0$$

3. Verfahren nach Anspruch 1,
dadurch gekennzeichnet, daß einer ersten Gruppe von Informationswörtern erste und zweite Codewörter zugeordnet sind, für die die Summe

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j$$

einen Wert +a bzw. -b besitzt, und beim Erscheinen eines Informationswortes der ersten Gruppe eine Wahl zwischen den beiden möglichen Codewörtern abhängig von der laufenden Summe

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

über alle vorangehenden Codewörter derart getroffen wird, daß die laufende Summe minimisiert wird.

4. Verfahren nach Anspruch 2,
dadurch gekennzeichnet, daß einer zweiten Gruppe von Informationswörtern erste und zweite Codewörter zugeordnet sind, für die die Summe

$$\sum_{k=1}^{n} \sum_{j=1}^{k} x_j$$

einen Wert +a bzw. -b besitzt, und beim Erscheinen eines Informationswortes der zweiten Gruppe eine Wahl zwischen den beiden möglichen Codewörtern abhängig von der laufenden Summe

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

über alle vorangehenden Codewörter derart getroffen wird, daß die laufende Summe minimisiert wird.

5. Verfahren nach Anspruch 3 oder 4,
dadurch gekennzeichnet, daß die Parameter a und b gleich sind.

6. Coder zur Verwendung mit dem Verfahren nach einem oder mehreren der vorangehenden Ansprüche,
dadurch gekennzeichnet, daß ein Umsetzer zum Umsetzen der m-Bit-Informationswörter in n-bit-Codewörter derart vorgesehen ist, daß die laufende Summe

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

über sämtliche vorangehende Summenwerte über die Bits der erzeugten Codewörter ebenfalls beschränkt bleibt.

7. Decoder zur Verwendung mit dem Verfahren nach einem oder mehreren der vorangehenden Ansprüche,
dadurch gekennzeichnet, daß ein Umsetzer zum Umsetzen der n-bit-Codewörter in m-Bit-Informationswörter derart vorgesehen ist, daß die laufende Summe

$$\sum_{k=1}^{i} \sum_{j=1}^{k} x_j$$

über sämtliche vorangehende Summenwerte über die Bits der empfangenen Codewörter ebenfalls beschränkt bleibt.

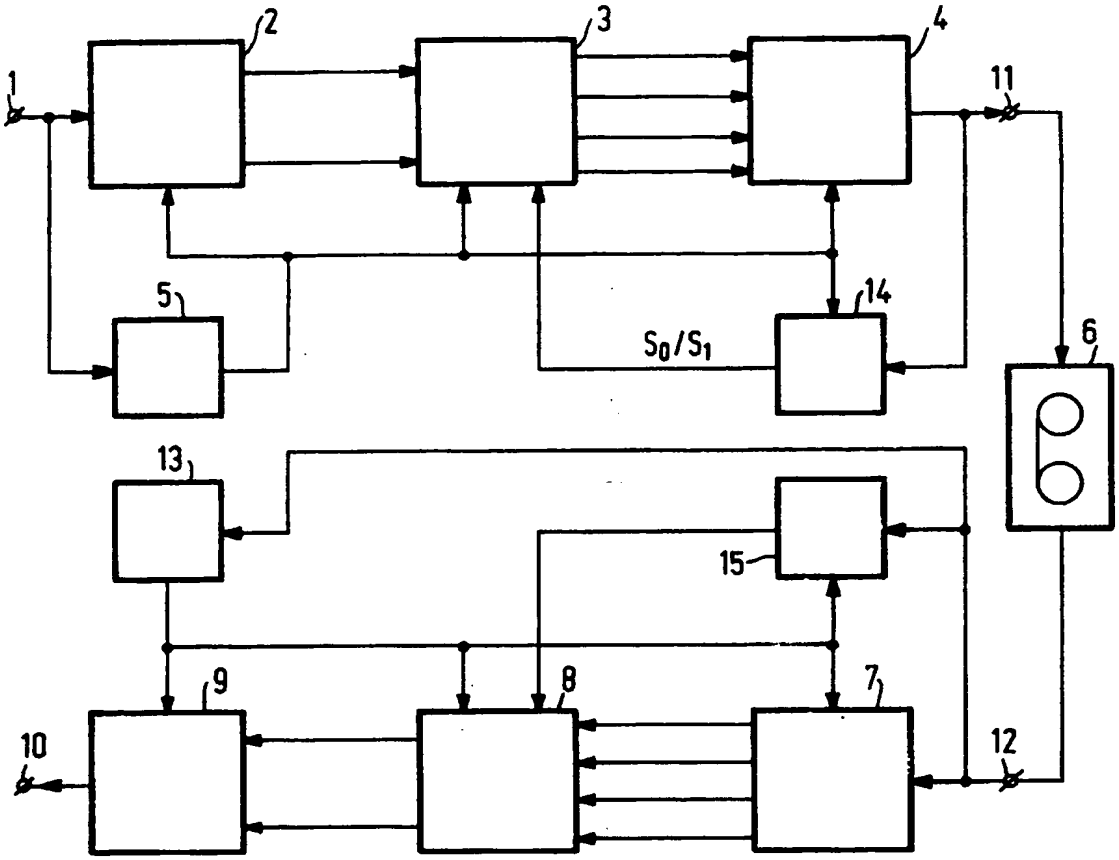

FIG.1

0 0	0 1 0 1
0 1	0 1 1 0
1 0	1 0 0 1
1 1	1 0 1 0

FIG.2

0	0 1 1 0
1	1 0 0 1

FIG.3

n	M	R
4	2	0.250
8	8	0.375
12	58	0.488
16	526	0.565
20	5448	0.621
24	61108	0.662
28	723354	0.695
32	8908546	0.722
36	113093022	0.743

FIG.4

r =		2	4	6	8	10
n =	4	.396	.5	–	–	–
	8	.488	.557	.594	.625	.641
	12	.568	.616	.648	.672	.689
	16	.627	.663	.688	.707	.722
	20	.670	.700	.720	.736	.748

FIG.5

	$z_1(i) < 0$	$z_1(n)$	$z_1(i) > 0$	$z_1(n)$
0 0	1 0 0 1	0	1 0 0 1	0
0 1	0 1 1 0	0	0 1 1 0	0
1 0	1 0 1 0	+2	0 1 0 1	−2
1 1	1 1 0 0	+4	0 0 1 1	−4

FIG.6

	$Z_1(i)$			$Z_1(n)$	$Z_1(i+4)$	
a_3 a_2	a_1 a_0		b_5 b_4 b_3 b_2		b_1 b_0	
0 0	0 0	(+3)	1 0 0 1	0	0 0	(+3)
0 0	0 1	(+1)	1 0 0 1	0	0 1	(+1)
0 0	1 0	(-1)	1 0 0 1	0	1 0	(-1)
0 0	1 1	(-3)	1 0 0 1	0	1 1	(-3)
0 1	0 0	(+3)	0 1 1 0	0	0 0	(+3)
0 1	0 1	(+1)	0 1 1 0	0	0 1	(+1)
0 1	1 0	(-1)	0 1 1 0	0	1 0	(-1)
0 1	1 1	(-3)	0 1 1 0	0	1 1	(-3)
1 0	0 0	(+3)	0 1 0 1	-2	0 1	(+1)
1 0	0 1	(+1)	0 1 0 1	-2	1 0	(-1)
1 0	1 0	(-1)	1 0 1 0	+2	0 1	(+1)
1 0	1 1	(-3)	1 0 1 0	+2	1 0	(-1)
1 1	0 0	(+3)	0 0 1 1	-4	1 0	(-1)
1 1	0 1	(+1)	0 0 1 1	-4	1 1	(-3)
1 1	1 0	(-1)	1 1 0 0	+4	0 0	(+3)
1 1	1 1	(-3)	1 1 0 0	+4	0 1	(+1)

FIG.7

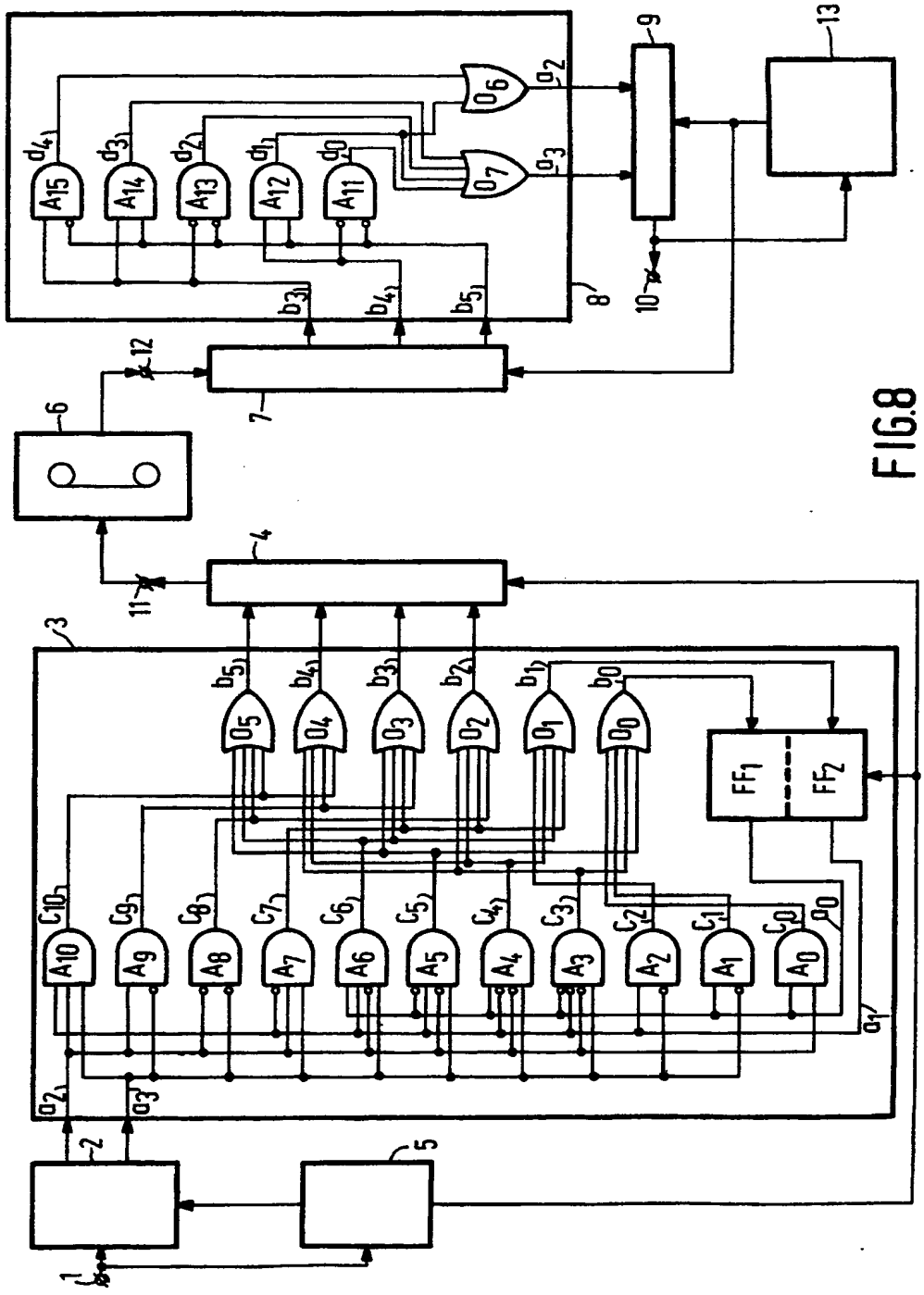
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
O6
O7
O5
O4
O3
O2
O1
O0
FF1
FF2
1
2
3
4
5
6
7
8
9
10
11
12
13

FIG8

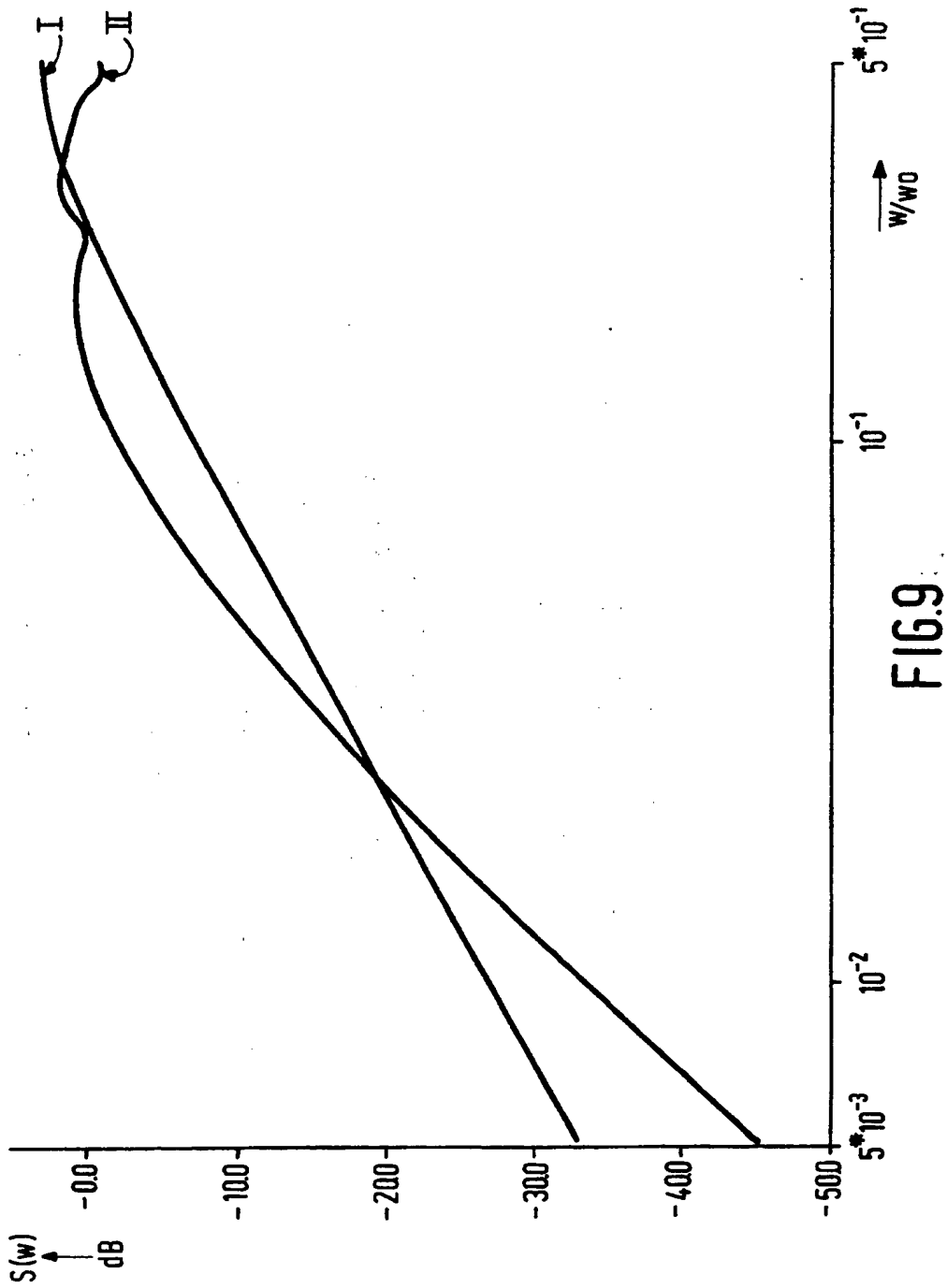

FIG.9